Activity-conditioned continuous human pose estimation for performance analysis of athletes using the example of swimming

Moritz Einfalt Dan Zecha Rainer Lienhart
Multimedia Computing and Computer Vision Lab, University of Augsburg
moritz.einfalt@informatik.uni-augsburg.de

Abstract

In this paper we consider the problem of human pose estimation in real-world videos of swimmers. Swimming channels allow filming swimmers simultaneously above and below the water surface with a single stationary camera. These recordings can be used to quantitatively assess the athletes' performance. The quantitative evaluation, so far, requires manual annotations of body parts in each video frame. We therefore apply the concept of CNNs in order to automatically infer the required pose information. Starting with an off-the-shelf architecture, we develop extensions to leverage activity information – in our case the swimming style of an athlete – and the continuous nature of the video recordings. Our main contributions are threefold: (a) We apply and evaluate a fine-tuned Convolutional Pose Machine architecture as a baseline in our very challenging aquatic environment and discuss its error modes, (b) we propose an extension to input swimming style information into the fully convolutional architecture and (c) modify the architecture for continuous pose estimation in videos. With these additions we achieve reliable pose estimates with up to $+16\%$ more correct body joint detections compared to the baseline architecture.

1. Introduction

In recent years, an increasing interest in computer vision applications in the sports domain can be observed. One important reason is that broadcasts of sport events are among the most popular content on TV and the internet [27]. The footage offers plenty of possibilities to gather additional statistics, including team performance statistics for the participating athletes and their coaches. In this work we focus on the specific scenario of video recordings of top-tier swimming athletes in a swimming channel. In competitive swimming, such swimming channels can be used for individual performance analysis. They consist of a pool with an adjustable artificial water current, flowing in a fixed direction. Cameras are positioned at various locations around the pool, both above and below the water surface, to record the athlete in the channel. By matching the flow velocity, the athlete can perform normal swimming motion while staying in the same position relative to the cameras. This enables the recording of swimming motion over a long period of time. The recordings are then used by an expert of the field to quantitatively asses the performance and to work out possible improvements of the athlete's technique and further training strategies. Inferring information about the athlete's movements such as the stroke rate or other intra-cyclic kinematic parameters over time requires to annotate the video material appropriately. These annotations can range from a sparse selection of frames showing characteristic key-poses to frame-by-frame locations of the athletes body parts and/or joints [33]. Automating this process by employing a vision-based pose estimation system can alleviate the massive overhead of manual pose annotations. Moreover, it opens up new kinds of individual training sup-

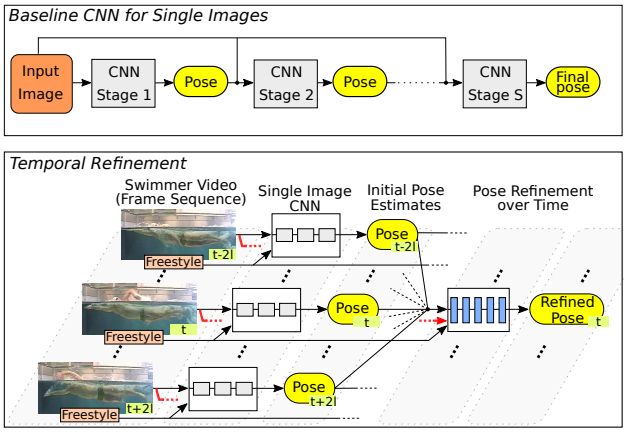


Figure 1. Overview of our approach to reliable human pose estimation in swimming channel videos. Top: Baseline CNN from [30] for human pose estimation on single images. Bottom: Our architecture uses initial estimates from the baseline CNN and refines these over fixed length pose sequences. Network layers are conditioned on the swimming style of an athlete.

port to even more athletes as much more regular quantitative data is available for performance and improvement analysis.

Human pose estimation is a long standing task in the domain of computer vision. There exists a large research community that is constantly striving to develop better and more reliable methods. With the continuing success of Convolutional Neural Networks (CNNs), several architectures for end-to-end human pose estimation have emerged, including the Convolutional Pose Machines (CPMs) of Wei et al. [30]. Based on their notable success and influence on other architectures [12, 16], we apply the CPM framework to our challenging scenario of human pose estimation in swimming channel recordings and analyze its performance and failure modes. Based on these, we propose architectural improvements to (a) leverage additional activity information including the swimming style of an athlete and to (b) exploit the temporal connection of poses in consecutive frames, i.e., the fact that poses are estimated in temporally and causally related video frames. Figure 1 gives an overview of our approach.

2. Related work

Computer vision has been adopted for various applications in the sports domain. Prominent tasks include sports type [8] and activity recognition [17, 27], tracking athletes and other objects of interest in videos [24, 31] and human pose estimation [6, 11]. [15] offer an overview of a wide range of application.

For performance analysis of individual athletes, [10] propose a method to facilitate speed and stride length estimation for runners based on hand-held camera recordings. Specific to an aquatic environment, [22] describe how swimmers can be tracked when filmed by a moving camera above the water surface. [33] discuss the identification of characteristic poses of swimmers. [28] present a CNN approach for automatic stroke-rate estimation in swimmer videos.

The traditional approach to human pose estimation are pictorial structures, where the human body is modeled as a collection of interacting parts [2, 7, 14, 21, 29]. The model describes the appearance of individual parts and the relationship between interacting parts in a probabilistic fashion. The goal is to find the most probable part configuration given an input image.

Recent literature focuses on methods using CNNs to overcome the drawbacks of hand-crafted image features and limited part interactions present in classical approaches. The currently best results on popular human pose estimation benchmarks like the Leeds Sports Pose (LSP) [14] and MPII Human Pose [1] datasets all apply CNNs. [26] describe an architecture that directly regresses the image coordinates of body joints. Subsequent publications regress confidence maps that indicate the likelihood of all possible joint locations in an image [11, 16, 18, 20]. This spatial encoding of the learning objective seems to be more natural to CNNs compared to the direct regression of image coordinates. Another common design are architectures performing iterative refinement [3–5]. After producing an initial pose estimate it is progressively refined in the deeper layers of the network. There are also proposals to use classical part-based models to refine the pose estimates from CNN-based methods, either as a separate post-processing step [19] or by mapping the domain-specific part interactions into the neural network itself for an end-to-end trainable architecture [32].

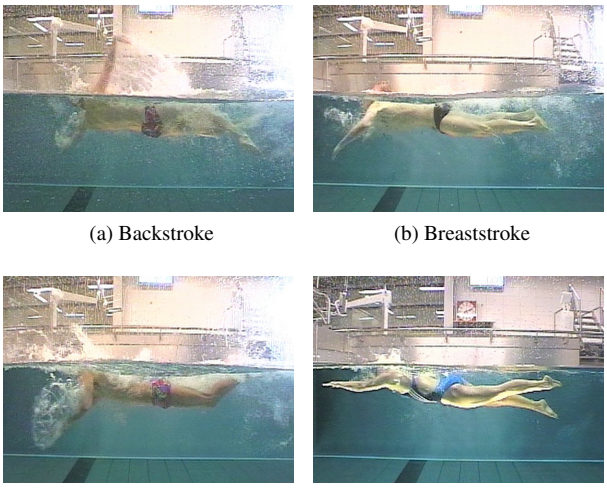
While most publications focus on human pose estimation on single 2D images, we are additionally interested in human pose estimation on videos. [6,34] use pictorial structures to model humans in videos. They extend the spatial interactions between body parts by temporal dependencies that describe the change of body part configurations over time. Flowing Conv-Nets [18] combine a CNN for human pose estimation on single images with a second CNN for the optical flow in videos that enables an estimate of the movement of body parts. In [23], optical flow and both spatial and temporal part interactions are used jointly in a single network architecture. [9] describe a recurrent neural network (RNN) architecture applied to sequential video frames. In our approach we avoid the computational expensive extraction of optical flow and the data-intensive training of RNNs due to limited video material.

3. Human pose estimation for swimming channel recordings

The video material used in this work has been recorded at $720 \times 576@50$i with a single stationary camera behind a glass pane at the left side of a swimming channel. The athletes are depicted from the side, partially above and below the water surface. The swimming direction is always right to left. The videos display different male and female athletes at different flow velocities performing four different swimming styles: backstroke, breaststroke, butterfly and freestyle. The swimming style throughout each video does not change. Figure 2 shows exemplary video frames for all four styles.

The video frames are annotated using a person-centric, 14 joint body model. For the symmetric swimming styles such as breaststroke and butterfly only the left side of the body is annotated by hand. Due to the side-view, the body parts on the right side are usually directly occluded by their left counterpart. Using the same image coordinates for both left and right joints is thus a good approximation in most cases. For backstroke and freestyle (anti-symmetrical motion) all 14 joints are annotated explicitly.

The viewpoint of the camera and the underwater setting present multiple challenges for human pose estimation:



(a) Backstroke (b) Breaststroke

(c) Butterfly (d) Freestyle

Figure 2. Exemplary frames from the swimming channel footage. The data poses multiple challenges: (a) Image noise due to bubbles and spray. (b) Ambiguous joint locations (head) due to refractions at the water surface. (c) Occlusion by the water line (head, lower legs). (d) Self-occlusion (left arm).

- Image noise due to bubbles and spray (Figure 2a).
- Ambiguous joint locations due to refraction at the water surface (Figure 2b).
- Occluded joints by the water surface (Figure 2c).
- Frequent self-occlusions (Figure 2d).

This makes precise joint localization difficult, even for a human observer. At the same time, the variety of poses, appearance and background is rather limited compared to unrestricted human pose estimation in the wild.

3.1. Baseline approach

As a baseline method we use the CPM framework from [30] for human pose estimation on single images. It uses a pure convolutional neural network divided into S identical stages. The network is trained on instances $(x, \mathbf{y})$, where x is the input RGB image of fixed size, centered on the person of interest. $\mathbf{y} = (y_1, \dots y_J)$ represents the ground truth locations of all J joints in Cartesian image coordinates. The objective of the network is to regress confidence values for all possible joint locations. The output of every stage $s \in [1, S]$ is a stack of confidence maps (in the following simply denoted as *heatmaps*) $\hat{\mathbf{h}}^s = (\hat{h}^s_1, \dots, \hat{h}^s_J)$. For every image location $z \in Z$, $\hat{h}^s_j(z)$ is interpreted as the confidence that joint j is located at z. The locations with highest confidence in the final set of heatmaps $\hat{\mathbf{h}}^S$ are used as the predicted locations $\hat{y}_j$ for each joint j:

$$\hat{y}_j = \arg\max_{z \in Z} \hat{h}^S_j(z). \quad (1)$$

The main motivation for using a deep, stage-wise CNN architecture is to utilize spatial context and learn dependencies between any set of joints. This is necessary for improved estimates on difficult instances, e.g. when some body parts are occluded or when distinguishing left-side body parts from their right-side counterparts. The first network stage uses the input image to predict a first set of joint heatmaps, which is then subsequently refined by the following stages (see Figure 1). The increasing receptive field and thus the additional spatial context with each subsequent stage enables the network to resolve ambiguities in the estimates of previous stages. In this work we consider a fine-tuned CPM with three stages, initialized on general human pose estimation in sports. It operates on each video frame individually and acts as the baseline architecture.

4. Utilizing swimming style information

A key challenge in our aquatic environment are frequently occluded body parts, either hidden behind other parts of the body or the waterline. Detecting these body parts is necessary to enable the evaluation of an athletes movement over time, *e.g.* for inferring kinematic parameters like angular and vectorial velocities. Despite the CNNs ability to implicitly learn joint and body part dependencies, frequent occlusions can make an unambiguous reconstruction of an athletes pose very difficult. Figure 2d shows an exemplary video frame, where the athletes left arm is not directly visible. Given only this single frame, even a human observer might not be able to correctly estimate the pose. However, the pose of a swimmer is directly related to his swimming style. Information about his swimming style can resolve the ambiguous location of the left arm, since *e.g.* the positioning of both arms on top of each other is unlikely for a freestyle swimmer. We therefore want to encode this additional contextual information into the CNN architecture as it further reduces the space of possible poses and enables the learning of more specific body part dependencies. In Section 5 we will also exploit the temporal aspect of motion to improve further.

4.1. One-hot class label maps

The additional swimming style information partitions the data into four distinct classes. Each video and thus each frame is labeled to belong to one of these classes. We propose a very simple way to encode class labels by using additional 2D input channels, one for each class, and call them *class label maps*. The class label maps are one-hot encoded, *i.e.*, the respective map of a given swimming style is filled with ones, while the remaining three maps are zeroed. These additional maps can be added as supplementary input channels to one or multiple convolution layers in the network. We choose this encoding to retain a pure convolutional architecture in favor of *e.g.* the addition of

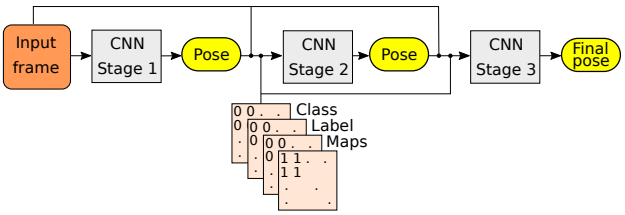


Figure 3. "Style input - once" architecture using one-hot encoded class label maps as additional input.

an encode-decoder network block where an abstract hidden state can be combined with arbitrary additional input information [25].

When adding class label maps to a convolutional layer, the new filter weights shift the filter activations by a constant but class dependent value (since the class label input is constant as well). This is identical to an additional class dependent bias term learned for each filter. With a subsequent ReLU activation function, the filter activations are either increased, decreased or set to zero when falling below the ReLU threshold. The observed effect is an on-off switching functionality: Activations of a filter surpass the threshold only if a specific subset of class labels is given. It enables the CNN to learn general purpose filters as well as swimming style specific ones and to combine them appropriately.

The benefit of swimming style information is that it reduces the variety of expected poses and enables learning of more powerful dependencies between joints. Adding the swimming style input at the beginning of the baseline architecture might thus not be useful, as the layers of the first network stage operate on local image content with a limited receptive field. The subsequent stages seem more appropriate, as they have the capability to learn long-range spatial dependencies between different joints and can naturally benefit from the additional input. We evaluate two strategies for adding swimming style class label maps to the baseline architecture. The first approach is to add class label maps once at the beginning of each stage $s > 2$, as depicted in Figure 3. The network has to propagate the class information manually if it is required in deeper layers. In the second approach, class label maps are added to all subsequent network layers as well, effectively replicating the class information. We denote the two variants "Style input - once" and "Style input - repeated".

5. Temporal pose refinement

For our baseline approach, we naively perform human pose estimation on single video frames, ignoring the temporal component of the video data. The same is true for the swimming style conditioned variant, where the only information shared across time is the constant swimming style in each video. While the pose of an athlete is not constant, the changes from one frame to another are limited. This is especially true for the swimmer videos recorded at 50Hz. Thus, there is a strong dependency between the pose in the current frame and poses in past and future frames. If past poses are known, they can act as an important cue for what pose to expect in the next frame(s).

Given the visual challenges in swimmer videos (see Figure 2), precise estimation of joint locations on single frames can be difficult. Sequential video frames enable humans to estimate the pose of athletes with higher precision by interpolating the location of joints (and thus their movement) over time. We propose an architectural extension to the baseline CNN that follows this idea of temporal refinement.

5.1. Temporal sequence model

The stage-wise architecture of the baseline CNN is intended to facilitate learning of spatial dependencies. In the same manner, one can imagine an additional network stage to learn temporal dependencies. We propose to exploit the temporal dependencies by a stage that uses the estimated poses on past and future video frames to improve the pose for the current one. To avoid explicit recurrence and enable pose refinement both forward and backward in time, this stage is a separate CNN operating on sequences of single-frame pose estimates. It acts as an additional post-processing network that refines pose estimates over time.

Our overall architecture for human pose estimation on videos consists of the baseline 3-stage CNN and the temporal post-processing network that we now describe. Each video consists of frames $(x_1, \ldots, x_T)$, where $T \in \mathbb{N}$ is the length the video. We apply the baseline CNN to each frame x_t, $t \in [1, T]$, and obtain a single-frame pose estimate in the form of localization heatmaps $\hat{\mathbf{h}}_t = (\hat{h}_{t,1}, \ldots, \hat{h}_{t,J})$ for the $J = 14$ joints. Our post-processing network uses frame x_t and the *sequence of pose estimates* $\mathbf{z}_t$ with

$$\mathbf{z}_t = (\hat{\mathbf{h}}_{t-2l}, \hat{\mathbf{h}}_{t-2(l-1)}, \ldots, \hat{\mathbf{h}}_{t+2l}) \tag{2}$$

as input. It outputs a refined estimate $\mathbf{h}_t^*$, which is again a set of J joint localization heatmaps for the frame at time t. The free parameter $l \in \mathbb{N}$ defines how many single-frame estimates from past and future frames are used to refine a pose. We denote the length of the sequence by k with $k = 4l + 1$. Note that we use pose sequences with reduced temporal resolution which contain pose estimates from every other frame only. This leads to an effective input sequence length of $k' = 2l + 1$ and enables long pose sequences for refinement while keeping the input size feasible.

5.2. Network architecture for temporal interpolation of joint predictions

The proposed temporal refinement network depicted in Figure 4 takes $(x_t, \mathbf{z}_t)$ as its input and is trained to predict

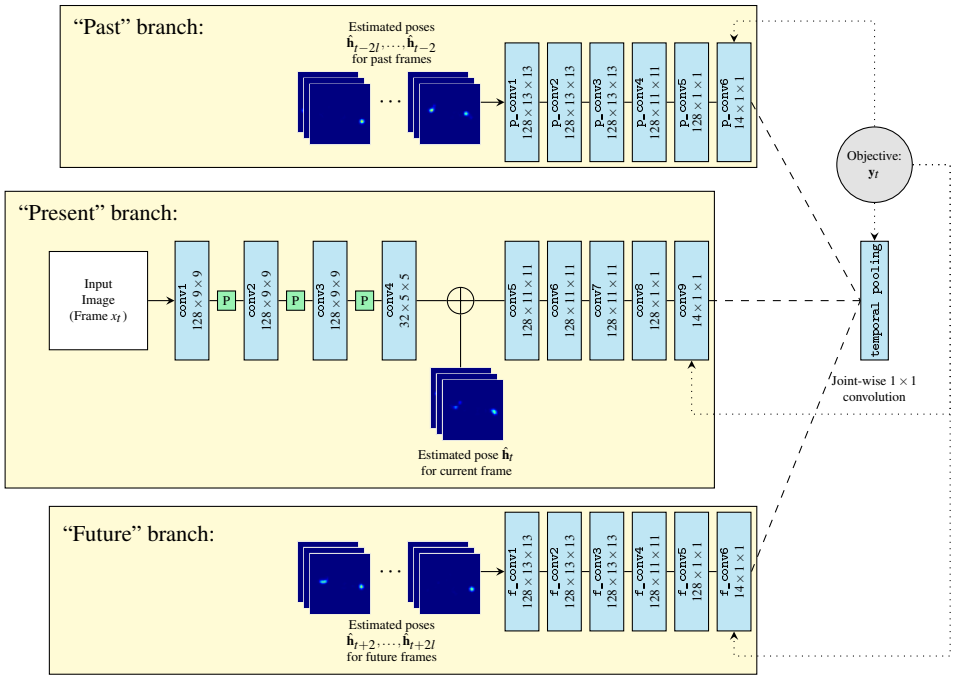


Figure 4. Network architecture for temporal pose refinement. Pose estimates from past and future frames are processed in separate network branches. Each branch is trained to predict heatmaps representing the pose $\mathbf{y}_t$ for the current frame x_t. The temporal pooling layer combines predicted heatmaps from all three branches and is trained separately. **P** denotes $\times 2$ max. pooling.

the ground truth joint location $\mathbf{y}_t$. However, the input is not simply stacked and processed jointly. The network architecture is designed to reflect the idea of temporal interpolation of joint locations based on predicted locations in surrounding video frames, *i.e.* based only on $\mathbf{z}_t$. We enforce this mechanism by dividing the network input into three parts: the single-frame pose estimates for past frames, the estimates for future frames and the estimate for the current frame together with the video frame itself. All three parts are processed in different network branches. Each branch is trained to predict the same target – the pose for the current frame. The "past" and "future" branches do not possess any image information and are thus forced to predict the current joint locations solely based on the joint estimates in the preceding or subsequent frames. This requires to infer some notion of interpolation forward or backward in time. The "present" branch does not have any temporal information and resembles a normal network stage in the baseline architecture. The outer branches use convolution layers with larger filter kernels for an increased receptive field.

The final layer of the network is intended to integrate the predictions of all three branches. It is based on the idea of *temporal pooling* in [18]: For each joint separately, a single 1x1 convolution filter is learned on the respective heatmaps from all three branches. Each filter computes a weighted average to combine the prediction heatmaps for one specific joint. It consists of only three weights, one for each branch, and aggregates the different predictions based on the past, present and future.

	Backstr.	Breaststr.	Butterfly	Freestyle
Training	1765	1464	1600	1200
Test	400	317	200	200

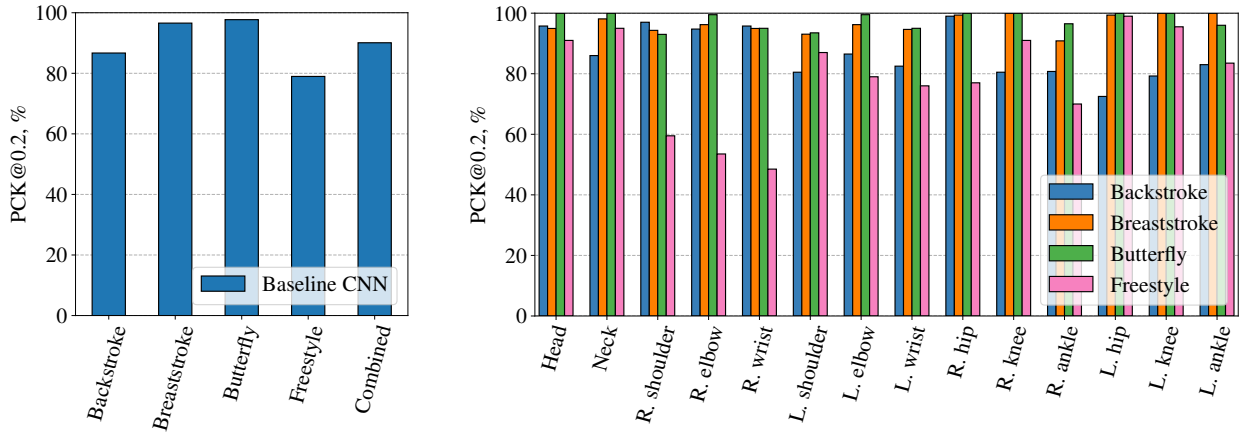
Table 1. Number of video frames in the training and test set.

6. Evaluation

We now evaluate the baseline architecture as well as the proposed additions for swimming style information and temporal refinement on the swimming channel data.

Data partitioning The swimming channel data consists of 24 videos with 200 to 400 frames each, leading to a total of 7146 annotated video frames. We partition the data into training and test sets. Due to the cyclic nature of swimming motion, similar poses with similar visual appearance occur multiple times in one video. Hence, the partitioning is performed on video boundaries such that frames from one video are either all in the training set or in the test set. One video per swimming style is held out as the test set, while the remaining videos are used for training. Training and test set sizes are depicted in Table 1.

Metric For a quantitative evaluation of pose estimates we apply the *Percentage of Correct Keypoints* (PCK) metric



(a) (b)

Figure 5. Results of the baseline CNN on the swimmer test set. PCK@0.2, (a) combined across all joints, (b) for individual joints.

[21]. The PCK metric counts a joint as correctly localized if the euclidean distance to the ground truth location does not exceed a fixed fraction α of the torso diameter, which is defined as the distance between the left side of the hip and the right shoulder. We use $\alpha = 0.2$ to compare experimental results.

6.1. Baseline results

We use a 3-stage version of the CNN architecture from [30] as a performance baseline. The network is initialized by training on the LSP dataset for general human pose estimation in sports. Afterwards, the network is fine-tuned on individual frames of the swimming channel data. All experiments are performed with Caffe [13] on a NVIDIA Titan Xp GPU.

The swimmer test set performance of the baseline approach is depicted in Figure 5a. We achieve a detection rate of 90.1% using the PCK@0.2 metric across the complete swimmer test set. Table 1 reveals that the four swimming styles are not represented equally in the test set. It contains twice as many examples for back- and breaststroke compared to freestyle. Hence, the PCK score on the complete test set is biased towards backstroke and breaststroke performance. We additionally report performance on each style separately in Figure 5a. The results on breaststroke and butterfly are excellent with 96.6% and 97.7%, respectively. For backstroke and freestyle, only 86.7% and 79.0% of joints are detected correctly. This indicates that symmetrical and anti-symmetrical swimming styles pose challenges of varying difficulty. Evidently, pose estimation for breaststroke and butterfly examples seems to be an easier task, since left and right joints share the same annotations.

Figure 5b depicts the joint-wise PCK for each swimming style. Reliable localization of head and neck is independent of the swimming style. Performance on breaststroke and butterfly is good across all joints ($> 90\%$). Hardly any difference between left and right joints can be observed here. Obviously, the CNN has learned to treat those two styles appropriately (no left-right differentiation). For backstroke, most errors occur on joints of the left arm and leg which are frequently occluded. This also affects the performance on the right ankle and wrist due to left-right confusion. Similar observations can be made for freestyle, where occlusion of the right parts of the body is frequent. Performance on the right arm is especially low, with a right wrist detection rate of 48.5%. Figure 8 shows some qualitative examples on the different swimming styles.

6.2. Effect of swimming style labels

To study the benefit of swimming style information we train the network variants with swimming style labels from Section 4.1 in the same manner as the baseline architecture with LSP-based initialization and fine-tuning on the swimmer data. New filter weights due to the added class label maps are initialized randomly. Figure 6 depicts the test set results. Both network variants achieve equal or better results compared to the baseline approach. We observe no significant change for backstroke and butterfly. This confirms that the baseline CNN can already recognize and handle these instances appropriately. For the anti-symmetrical strokes, the explicit swimming style information leads to a notable improvement. Repeated addition of the swimming style labels to subsequent convolution layers proves to be the superior approach for freestyle, leading to a gain of $+12.6$ PCK. The "Style input - once " variant however slightly dominates for the other strokes. Detailed results are listed in Table 3. Both variants reach close to equal performance on the combined test set. The results verify that the swimming style (or information about a persons movement or activity in general) is a beneficial supplementary in-

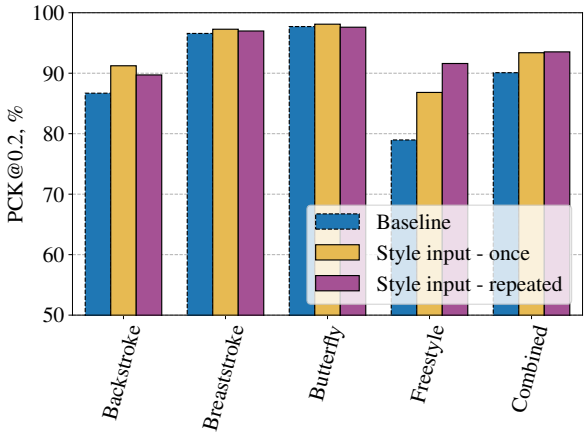


Figure 6. Test set results of the network variants with swimming style input. For convenience, baseline results are shown again (stroked).

formation that can improve human pose estimation performance significantly. Our proposed encoding of class labels with spatially replicated one-hot class label maps is confirmed as a viable approach for categorial inputs in CNNs.

6.3. Pose refinement over time

Finally, we evaluate our temporal pose refinement network from Section 5.2. We want to analyze the effect of the refinement mechanism alone and thus use no swimming style information for now. The network is therefore trained and evaluated using single-frame pose estimates obtained by the baseline CNN. We split the training up in two phases: First, each network branch is trained separately to predict the pose for the current frame. The final temporal pooling layer is ignored. In the second phase, this last layer is trained exclusively. We use this training scheme to force each branch to predict the same target.

We expect that the sequence length parameter k has a direct influence on pose estimation (refinement) performance. We evaluate multiple values for $k \in [1, 41]$ and show the results in Figure 7. The case $k = 1$ can be seen as a verification experiment. With no sequential information at all, the outer network branches can be ignored and we only have a fourth network stage in the baseline CNN architecture. [30] argue that additional stages up to a total of $s = 6$ are beneficial, at least when evaluated on the LSP dataset. The results with $k = 1$ reach and slightly surpass the baseline performance. They confirm that an additional and separately trained stage even without any past or future pose input is a viable option. When the network does use sequential pose estimates, *i.e.* $k > 1$, performance increases together with the sequence length. For $k = 29$, we achieve 93.8% on the combined test set, an additional $+1.9$ PCK compared to the baseline result. This improvement is almost entirely gained at backstroke and freestyle, with $+6.0$ PCK each. With even

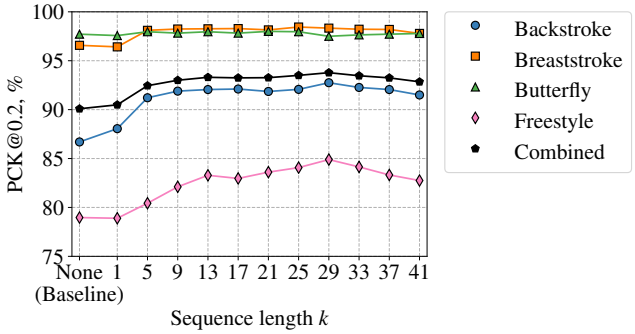


Figure 7. Effect of the sequence length k on the temporal refinement network, evaluated on the swimmer test set. Input sequences are of length 1 to 41, but include pose estimates from every second frame only.

	p_conv6	conv9	f_conv6
Avg. weight	0.289	0.300	0.306

Table 2. Temporal pooling calculates a joint-wise weighted average on the output layers of the three network branches. The table shows the weights averaged over all joints.

longer input sequences, performance starts to decline again for some of the swimming styles. Additionally, we obseve that the increasingly larger network input renders the training less stable.

With the simple design of the temporal pooling layer at the end of the refinement network, its weights can be analyzed to see what influence each network branch has on the final output. The layer consists of 3 weights (and one bias term) for each joint, and computes a weighted average on the past, present and future prediction heatmaps. Table 2 depicts the layer weights averaged over all joints. Similar importance is assigned to all three branches. Consequently, the predictions based on past and future information (*i.e.* based on $\mathbf{z}_t$) are vital for the quality of the final network prediction – a design goal of this architecture.

6.4. Combined architecture

It is straightforward to combine the findings of the preceding evaluation into a single architecture with swimming style information and temporal refinement. We simply combine the "Style input - repeated" architecture from Section 4.1 for single-frame pose estimates and the temporal refinement network (identically augmented with swimming style information). The test set result is listed in Table 3. In this configuration we can further boost the pose estimation performance to excellent 95.7% on the complete test set. It shows that the swimming style and the joint predictions over time are both very relevant and complementary sources of information. Figure 8 depicts a comparison to the baseline results on some qualitative examples with different swimming styles. We additionally provide the results

	Backstroke	Breaststroke	Butterfly	Freestyle	Combined
Baseline CNN [30]	86.7	96.6	97.7	79.0	90.1
Style input - once	91.2	97.3	**98.1**	86.8	93.4
Style input - repeated	89.7	97.0	97.6	91.6	93.5
Temporal pose refinement, $k = 29$	**92.7**	98.3	97.5	85.0	93.8
Temporal pose refinement, $k = 29$, +swimming style	92.4	**98.5**	97.7	**95.7**	**95.7**

Table 3. Summary of the PCK@0.2 scores on the swimmer test set.

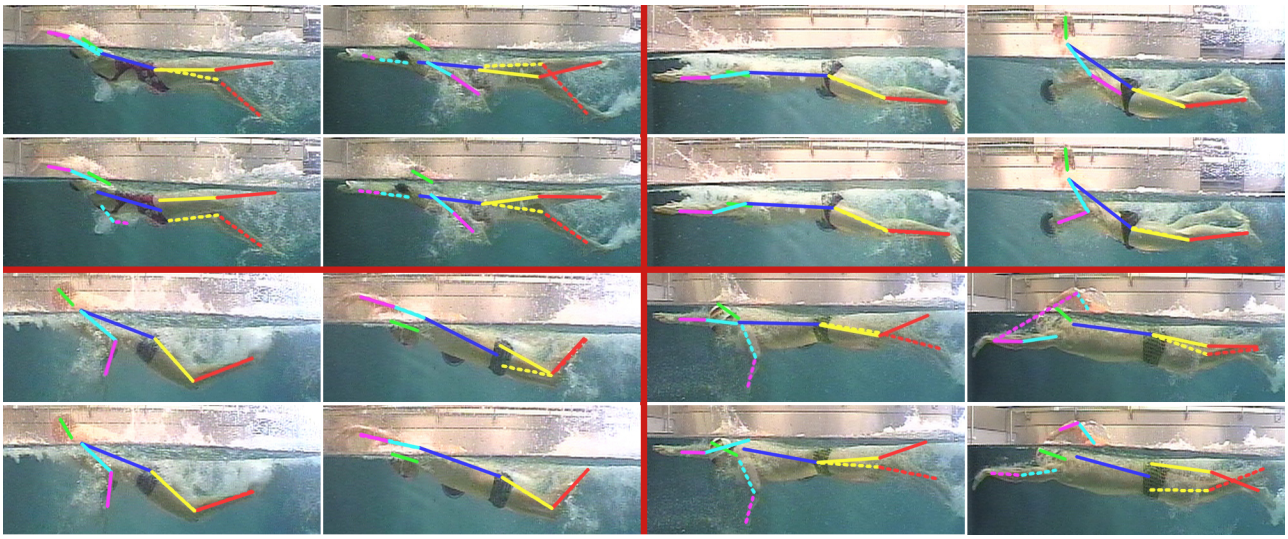


Figure 8. Qualitative results on the swimming channel data. The pose is visualized by drawing connections between respective joints. The parts on the left side of the body are stroked. Each quadrant contains examples from one swimming style: backstroke, breaststroke, freestyle, butterfly, from top left in clockwise order. Odd rows depict results from the baseline CNN, even rows depict the results on the same instances using our best architecture with temporal pose refinement and swimming style information.

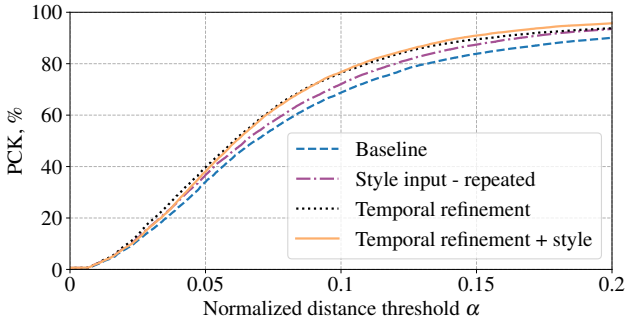


Figure 9. Results of all major network variants with a PCK threshold range $\alpha \in [0, 0.2]$

of our major network variants with varying PCK distance thresholds in Figure 9. It shows that our approaches can equally improve performance for higher precision joint detection.

7. Conclusion

This paper presents two extensions to the standard Convolutional Pose Machine model to deal with the challenges of human pose estimation for swimmers in the water. The first extension is based on the reduction of the subspace of possible pose configurations with swimmming style information. We formulate a spatially redundant one-hot encoding of class labels that allows the network to learn swimming style specific filters. This principle can be applied to any from of activity information. The second extension focuses on the temporal aspect of swimmer videos. We present a two-step approach where initial pose estimates are refined in fixed-length sequences by a separate CNN module. The results show that the network benefits from long sequences, indicating its ability to predict and refine poses forward and backward in time. Both extensions can be combined easily. We show that we clearly improve over the baseline CPM architecture. Our findings are directly applicable to other sports where reliable pose esimates are essential for effective performance analysis.

Acknowledgements

This work was funded by the Federal Institute for Sports Science based on a resolution of the German Bundestag. We would like to thank the Institute for Applied Training Science (IAT) Leipzig for providing the video data.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021, June 2009.

[3] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 468–475, May 2017.

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] M. Fastovets, J.-Y. Guillemaut, and A. Hilton. Athlete pose estimation from monocular tv sports footage. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[8] R. Gade and T. B. Moeslund. Sports type classification using signature heatmaps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2013.

[9] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *European Conference on Computer Vision*, pages 728–743. Springer, 2016.

[10] K. Hasegawa and H. Saito. Synthesis of a stroboscopic image from a hand-held camera sequence for a sports analysis. *Computational Visual Media*, 2(3):277–289, 2016.

[11] J. Hwang, S. Park, and N. Kwak. Athlete pose estimation by a global-local network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[12] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[14] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010.

[15] T. B. Moeslund, G. Thomas, and A. Hilton. *Computer vision in sports*. Springer, 2015.

[16] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[17] A. Nibali, Z. He, S. Morgan, and D. Greenwood. Extraction and classification of diving clips from continuous video footage. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[18] T. Pfister, J. Charles, and A. Zisserman. Flowing Conv–Nets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[19] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision*, pages 33–47. Springer, 2014.

[21] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3681, 2013.

[22] L. Sha, P. Lucey, S. Morgan, D. Pease, and S. Sridharan. Swimmer localization from a moving camera. In *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, Nov 2013.

[23] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] K. Soomro, S. Khokhar, and M. Shah. Tracking when the camera looks away. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[25] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3D models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.

[26] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[27] F. Turchini, L. Seidenari, and A. Del Bimbo. Understanding sport activities from correspondences of clustered trajectories. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[28] B. Victor, Z. He, S. Morgan, and D. Miniutti. Continuous video to simple signals for swimming stroke detection with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[29] F. Wang and Y. Li. Beyond physical connections: Tree models in human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[31] X. Wei, L. Sha, P. Lucey, P. Carr, S. Sridharan, and I. Matthews. Predicting ball ownership in basketball from a monocular view using only player trajectories. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[32] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[33] D. Zecha and R. Lienhart. Key-pose prediction in cyclic human motion. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 86–93. IEEE, 2015.

[34] D. Zhang and M. Shah. Human pose estimation in videos. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.